WHAT IS CLAIMED IS:

1. A method of error recovery in a lockstep computer processing system, the system comprising a primary processor and a secondary processor and a bridge to a network, comprising the steps of:

operating the primary and secondary processors in lockstep;

receiving an error notification resulting from an error in either the primary processor or the secondary processor;

determining if the error is a recoverable error; and

if the error is a recoverable error, then

saving the state of either the primary or the secondary processor to a memory; and

resetting and restarting the primary and secondary processors using the saved state.

2. The method of claim 1 further comprising the steps of:

detecting a divergence in the operation of the primary and secondary processors before receiving the error notification;

waiting for a predetermined time after detecting the divergence; and

if the error notification is received before the expiry of the predetermined time and if the error is determined to be a recoverable error, then treating the error as a recoverable error.

3. The method of claim 2 wherein, if the error notification is received after the expiry of the predetermined time, then treating the error as a non–recoverable error.

4. The method of claim 1 wherein a non-recoverable error on the secondary processor is treated as a recoverable error.

5. The method of claim 1 further comprising the steps of:

if the error is determined to be a non–recoverable error, then

disabling the bridge to the network before data corruption resulting from the error can propagate onto the network.

6. The method of claim 1 wherein a hardware error that results in the loss of a resource that is currently not being used by the primary processor is treated as a recoverable error.

7. The method of claim 1 wherein the error notification reports an error occurring in a hardware resource, and wherein the error notification includes an identifier that can be used to determine whether the hardware resource is critical or non–critical.

8. The method of claim 7 wherein the hardware resource is disabled if the hardware resource is non–critical.

9. The method of claim 8 wherein the hardware resource is retried after processor restart to determine if the error in the hardware resource can be cured by a processor reset.

10. The method of claim 2 wherein the system includes a single main memory, the step of detecting divergence comprises the steps of:

comparing memory commands generated by the primary processor with memory commands generated by the secondary processor;

executing only the memory commands generated by the primary processor; and

signaling a divergence detection if the memory commands issued by the primary processor differ from the memory commands issued by the secondary processor.

11. The method of claim 1 further comprising the steps of:

detecting a divergence in the operation of the primary and secondary processors at the bridge to the network; and

shutting off the bridge to the network immediately unless the error has previously been determined to be a recoverable error.

12. The method of claim 2 wherein the divergence detection is conducted by comparing unique signatures of processor state received from the primary and secondary processors.

13. The method of claim 12 wherein the unique signatures are generated by applying an algorithm to state information for the primary and secondary processors.

14. The method of claim 1 further comprising the steps of:

conducting first and second flushes of cache memory of either the primary or the secondary processor.

15. The method of claim 1 further comprising the steps of:

conducting a high–speed reset and restart of the bridge to the network.

16. The method of claim 15 wherein the bridge to the network has a custom high–speed reset and restart procedure.

17. The method of claim 1 further comprising the steps of:

setting a watchdog timer; and

treating the error as a non–recoverable error if the watchdog timer expires before the resetting of the primary and secondary processors.

18. The method of claim 17 wherein the step of treating the error as a non–recoverable error comprises the step of:

conducting a hard–reset of the lockstep computer processing system.

19. The method of claim 1 wherein the step of restarting the primary and secondary processors using the saved state further includes the step of:

running the bridge to the network from a main memory until a bridge local memory has been initialized.

20. The method of claim 1 wherein the lockstep computer processing system is being utilized by a network resource, the network resource:

sending a data message to the lockstep computer processing system, the data message being lost due to the resetting and restarting of the primary and secondary processors;

sending a first inquiry message to the lockstep computer processing system after a first timeout period, the first inquiry message being lost due to the lockstep computer processing system being unavailable; and

sending a second inquiry message after a second timeout period;

wherein the sum of the first and second timeout periods is selected to be greater than an expected recovery time for the lockstep computer processing system.

21. The system of claim 20 wherein the network resource sends out no retries of the data message until a response is received to an inquiry message.

22. A computer system comprising:

a primary processor and a secondary processor being configured to operate in lockstep; and

an error–handling module to receive an error notification resulting from an error in either the primary processor or the secondary processor, to determine if the error is a recoverable error, and, if the error is a recoverable error, then to initiate saving the state of either the primary or the secondary processor to a memory and to reset and restart the primary and secondary processors using the saved state.

23. The system of claim 22 wherein the error–handling module is further configured to receive a notification of a divergence in the operation of the primary and secondary processors before receiving the error notification, the error–handling module being further configured to

wait for a predetermined time after receiving the notification of divergence; and

if the error notification is received before the expiry of the predetermined time and if the error is determined to be a recoverable error, to treat the error as a recoverable error.

24. The system of claim 23 wherein, if the error notification is received after the expiry of the predetermined time, then the error–handling module treats the error as a non–recoverable error.

25. The system of claim 22 wherein a non-recoverable error on the secondary processor is treated as a recoverable error.

26. The system of claim 22 further comprising a bridge to a network, wherein

if the error is determined to be a non–recoverable error, then the system is configured to disable the bridge to the network before data corruption resulting from the error can propagate onto the network.

27. The system of claim 22 wherein, in use, a hardware error that results in a loss of a resource that is not being used by the primary processor is treated as a recoverable error.

28. The system of claim 22 wherein, in use, the error notification reports an error occurring in a hardware resource, and wherein the error notification includes an identifier that can be used to determine whether the hardware resource is critical or non–critical.

29. The system of claim 28 wherein the system is further configured to disable the hardware resource if the hardware resource is non–critical.

30. The system of claim 28 wherein the system is further configured to retry the hardware resource after processor restart to determine if the error in the hardware resource can be cured by a processor reset.

31. The system of claim 23 further comprising a main memory, the system being configured to detect divergence by:

comparing memory commands generated by the primary processor with memory commands generated by the secondary processor;

executing only the memory commands generated by the primary processor; and

signaling a divergence detection if the memory commands issued by the primary processor differ from the memory commands issued by the secondary processor.

32. The system of claim 22 further comprising:

a bridge to an external network, the computer system being configured to:

detect a divergence in the operation of the primary and secondary processors at the bridge to the network; and

shut off the bridge to the network immediately unless the error has previously been determined to be a recoverable error.

33. The system of claim 23 wherein the error–handling module does divergence detection by comparing unique signatures of processor state received from the primary and secondary processors.

34. The system of claim 32 wherein the unique signatures are generated by applying an algorithm to state information for the primary and secondary processors.

35. The system of claim 22 wherein the reset and restart of the primary and secondary processors includes the step of:

conducting first and second flushes of cache memory of either the primary or the secondary processor.

36. The system of claim 32 wherein the bridge is configured to conduct a high–speed reset and restart during the reset and restart of the primary and secondary processors.

37. The system of claim 36 wherein the bridge to the network has a custom high–speed reset and restart procedure.

38. The system of claim 22 further comprising a watchdog timer, the system treating the error as a non–recoverable error if the watchdog timer expires during the reset and restart of the primary and secondary processors.

39. The system of claim 38 wherein the system conducts a hard–reset of the lockstep computer processing system upon expiry of the watchdog timer.

40. The system of claim 22 further comprising:

a bridge to an external network, the computer system being configured run the bridge to the network from a main memory until a bridge local memory has been initialized upon the reset and restart of the primary and secondary processors.

41. The system of claim 22 further comprising a network bridge associated with the primary and secondary processor and a network resource for utilizing the primary and secondary processor over a network, the network resource being configured to:

send a data message to the network bridge over the network, and, when the data message is lost due to the resetting and restarting of the primary and secondary processors, to:

send a first inquiry message to the network bridge after a first timeout period, and, when the first inquiry message is lost, to:

send a second inquiry message after a second timeout period;

wherein the sum of the first and second timeout periods is selected to be greater than an expected recovery time for the primary and secondary processors.

42. The system of claim 41 wherein the network resource sends out no retries of the data message until a response is received to an inquiry message.